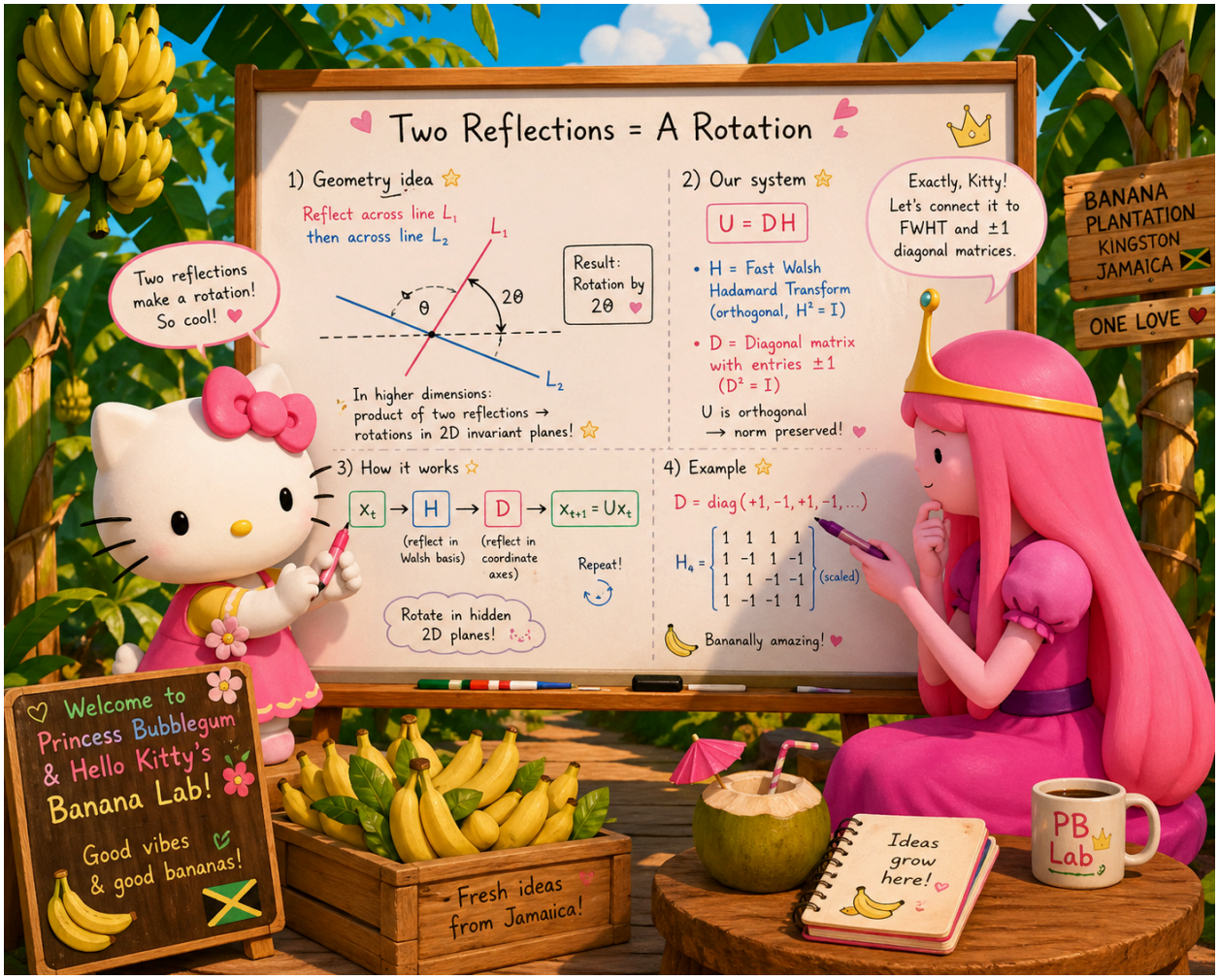
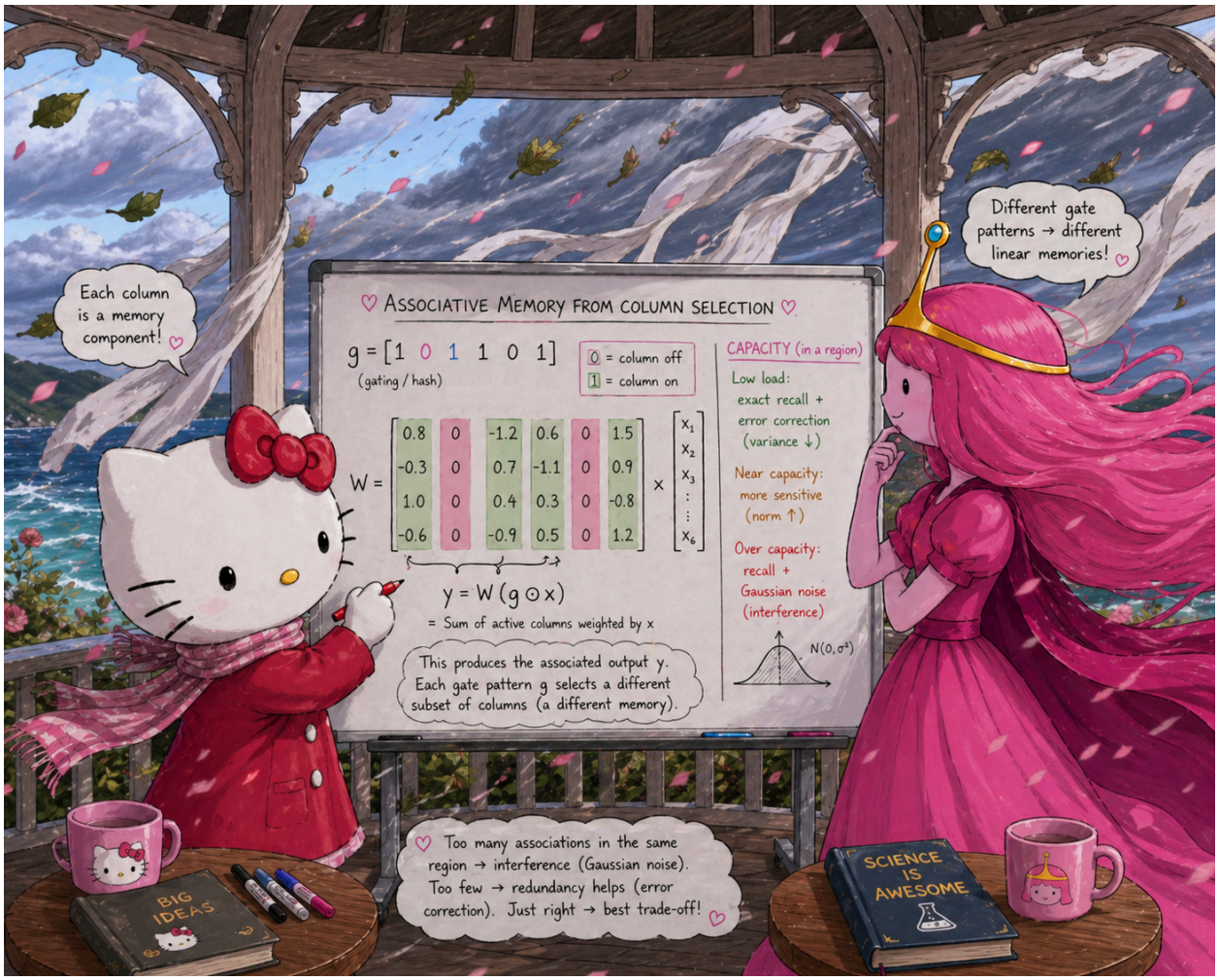
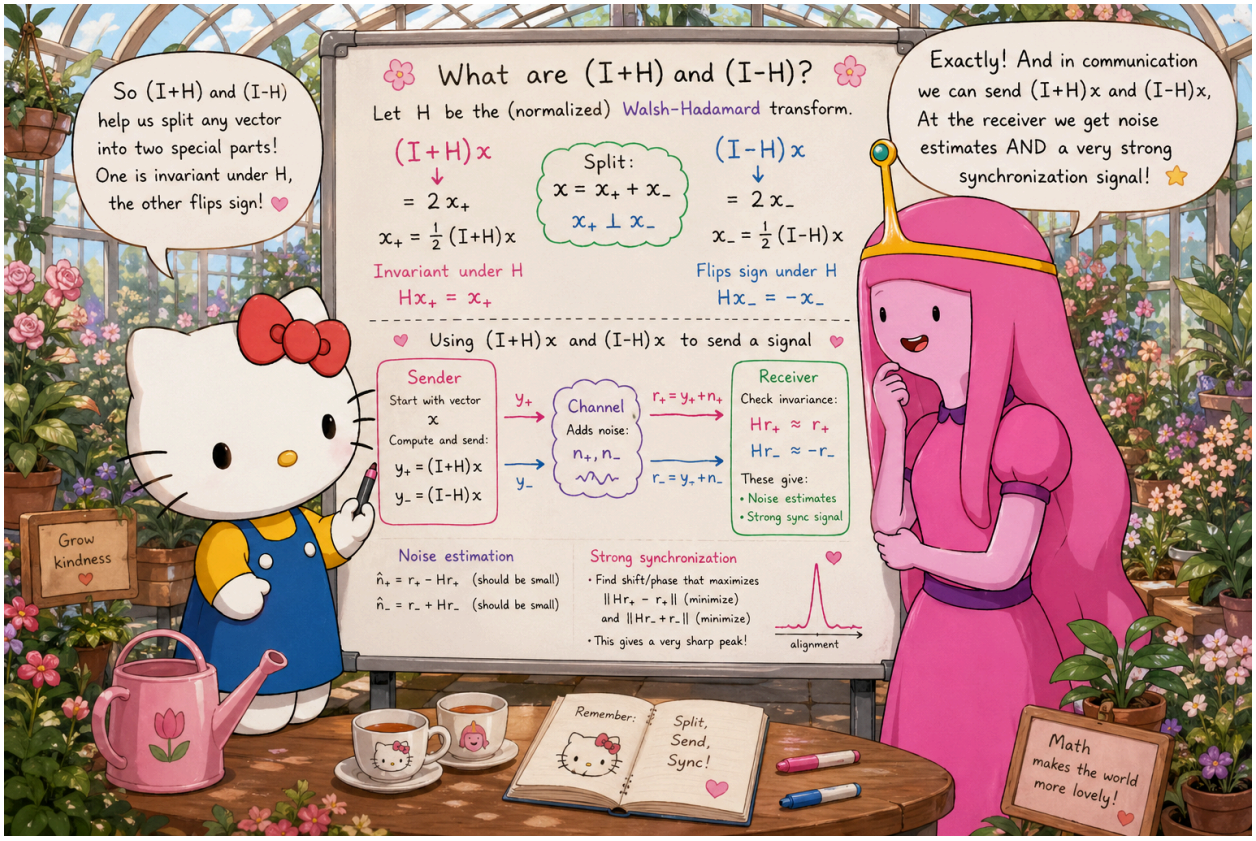
Pictorial Summaries:


Two Reflections = A Rotation
1) Geometry idea
Reflect across line L_1 then across line L_2
L_1
L_2
θ
2θ
Result: Rotation by 2θ
In higher dimensions: product of two reflections → rotations in 2D invariant planes!
2) Our system
U = DH
• H = Fast Walsh Hadamard Transform (orthogonal, $H^2 = I$)
• D = Diagonal matrix with entries ±1 ($D^2 = I$)
U is orthogonal → norm preserved!
3) How it works
x_t → H → D → $x_{t+1} = Ux_t$
(reflect in Walsh basis)
(reflect in coordinate axes)
Repeat!
Rotate in hidden 2D planes!
4) Example
$D = diag(+1, -1, +1, -1, \ldots)$
$H_4 = \begin{Bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{Bmatrix}$ (scaled)
Bananally amazing!
Two reflections make a rotation! So cool!
Exactly, Kitty! Let's connect it to FWHT and ±1 diagonal matrices.
BANANA PLANTATION KINGSTON JAMAICA
ONE LOVE
Welcome to Princess Bubblegum & Hello Kitty's Banana Lab!
Good vibes & good bananas!
Fresh ideas from Jamaica!
Ideas grow here!
PB Lab

Two Reflections Make a Rotation.


Each column is a memory component!
Different gate patterns → different linear memories!
♡ Associative Memory from column selection ♡
g = [1 0 1 1 0 1]
(gating / hash)
0 = column off
1 = column on
W =
0.8 0 -1.2 0.6 0 1.5
-0.3 0 0.7 -1.1 0 0.9
1.0 0 0.4 0.3 0 -0.8
-0.6 0 -0.9 0.5 0 1.2
x
x1 x2 x3 ⋮ x6
y = W (g ⊙ x)
= Sum of active columns weighted by x
This produces the associated output y. Each gate pattern g selects a different subset of columns (a different memory).
CAPACITY (in a region)
Low load: exact recall + error correction (variance ↓)
Near capacity: more sensitive (norm ↑)
Over capacity: recall + Gaussian noise (interference)
N(0, σ²)
♡ Too many associations in the same region → interference (Gaussian noise). Too few → redundancy helps (error correction). Just right → best trade-off! ♡
BIG IDEAS
SCIENCE IS AWESOME

Column selection in the next weight matrix from ReLU decisions in the current layer.
The next weight matrix will be later decimated by row selection from its own ReLU decisions. In terms of the final output of the network only about 25% of the parameters in the middle weight matrices are chosen to construct the output. However the selection process uses more than that (≈50%.)


So (I+H) and (I-H) help us split any vector into two special parts! One is invariant under H, the other flips sign!
What are (I+H) and (I-H)?
Let H be the (normalized) Walsh-Hadamard transform.
(I+H)x
= 2 x+
x+ = ½ (I+H)x
Split:
x = x+ + x-
x+ ⊥ x-
(I-H)x
= 2 x-
x- = ½ (I-H)x
Invariant under H
Hx+ = x+
Flips sign under H
Hx- = -x-
Using (I+H)x and (I-H)x to send a signal
Sender
Start with vector x
Compute and send:
y+ = (I+H)x
y- = (I-H)x
y+
y-
Channel
Adds noise:
n+, n-
r+ = y+ + n+
r- = y+ + n-
Receiver
Check invariance:
Hr+ ≈ r+
Hr- ≈ -r-
These give:
• Noise estimates
• Strong sync signal
Noise estimation
n̂+ = r+ - Hr+ (should be small)
n̂- = r- + Hr- (should be small)
Strong synchronization
• Find shift/phase that maximizes
||Hr+ - r+|| (minimize)
and ||Hr- + r-|| (minimize)
• This gives a very sharp peak!
alignment
Exactly! And in communication we can send (I+H)x and (I-H)x, At the receiver we get noise estimates AND a very strong synchronization signal!
Grow kindness
Remember:
Split, Send, Sync!
Math makes the world more lovely!

The eigenspaces of the fast Walsh Hadamard Transform.